NEURAL NETWORK IMPLEMENTATION FOR CHARACTER RECOGNITION

Prof..Sachin Patel
M.Tech(IT),Ph.D Pursuing
HOD(Information Technology)
PCST,Indore

Prof..Rakesh Pandit
M.Tech(IT),M.Phil(Comp.Sc.)
Assistant Professor
PCST,Indore

Mr.R. K. Rajbhure
M.Tech (Final Year, I T)
RGPV University
PCST,Indore.

***Abstract*- This paper describes a NEURAL NETWORK based technique for feature extraction applicable to segmentation-based word recognition systems. The proposed system extracts the geometric features of the character contour. The system gives a feature vector as its output. The feature vectors so generated from a training set, were then used to train a pattern recognition engine based on Neural Networks so that the system can be benchmarked. Since, an attempt was made to develop a system that used the methods that humans use to perceive handwritten characters. Hence a system that recognizes handwritten characters using Pattern recognition was developed.Here the data generated by comparison of two images was stored in excel format and then calling that data as an indivual input for generation of simulink diagram.**

Pattern recognition can be used to model human perception. The mathematics that Pattern recognition requires is extremely fundamental. Thus, any algorithm developed using Pattern recognition would require relatively simple and short calculations. Due to simplicity of calculations, they can be implemented on any hardware or software platform without too much concern for computing power. In this paper first part is about introduction to character Recognition. Then next part giving short introduction to Neuarl network implementaion for image processing using MATLAB.

Key terms: **Pattern recognition, image processing, handwritten character recognition, Euclidean distance, nearest neighbour algorithm, database, HCR, MATLAB Commands:bwmorph,imdilate,reshape,strlen,xor.**

OBJECTIVE

The main objective of handwritten character recognition is to interpret the contents of the data and to generate a description of that interpretation in the desired format. The objectives in the development of this method were, the recognition algorithm used here is very efficient. The goal of handwritten recognition is to interpret the contents of the data and to generate description of that interpretation in the desired format with greater accuracy. It also being used a paradigm to use it in future application to work with recognition of character with deaf and dumb people.

OVERVIEW

The overview of the process Iis mainly divided into three main processes and are shown in the figure 1.The first is the view process called as representation process where giving the input as a character is to get an image of the character and then treated in different ways to achieve a higher level form of the data. First, the image should undergo some image enhancements such as cropping, reshaping, and filtering out noise, this is called image pre-processing. The raw digitized data is then mapped to a higher level by extracting special characteristics and patterns of the image. This is called feature extraction. The higher level image is then stored in some special way, perhaps in a vector.

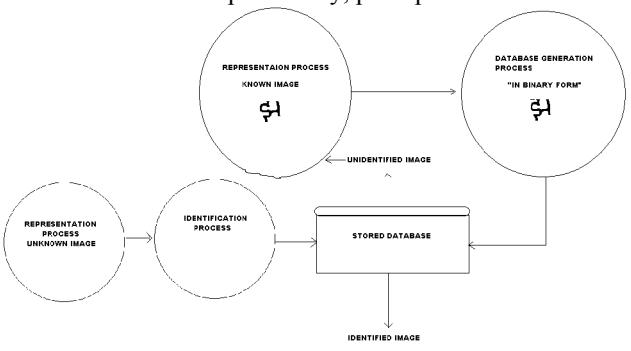


Fig.1: Overview of handwritten character recognition

The second process is the manipulation process called as generation of a known database containing the high level representation of all known characters (done in the same way as in the representation process).

This is the learning step where the system learns to separate the classes. The database is for knowing how all the different characters could be written and should contain quite a lot of different characters.

In the third part, the verification of the data with main database i.e.the identification/classification process classifies the unknown character given its high level representation and the information from the known database. There are several different approaches to make the identification. Here we use statistical method. The statistical approach is based on a similarity measure

I.INTRODUCTION

There are two distinct areas of research concerning Handwritten Character Recognition
(1) Off-line Character recognition, and
(2) On-line Character Recognition.
Here we use Off-line method. The Off-line character description data provides much more information for recognition. This is due to the reason that these images are usually of characters that were written earlier and later converted to digital format using a digital scanner. The challenge with Off-line character recognition is the development of a system that can recognize these characters.
This requires a system that requires very simple and short calculations. If not, the time taken to recognize the characters will render the system useless. Thus, the method selected for this was Pattern recognition.

In this, we present the handwritten character as part from common application Character and of recognizing the alphanumeric characters. The feature extraction algorithm applied to the characters is novel and leads to a very fast recognition. Although a number of good recognition algorithms have been proposed for handwritten character recognition, the achieved performance is still far from those of human beings in context free handwritten character recognition. The major obstacles have been the different handwriting styles and changeable writing conditions. These two aspects make handwritten characters extremely variable. Observing that the different writing styles or individual writing style of each writer is very important problem in handwritten recognition.

The handwritten identification requires the use of optical mouse as a scanner application for off-line conversion. It is the examination of the design, shape and structure of handwritten character. In this handwriting identification, writer are required to write the same fixed text or also called text-dependent. In practice the requirement for the use of fixed text takes writer identification prone to forgery.

In this technique to determine whether the character wrote is match with the character of the database was presented. This technique is based on computing the main Euclidean distances between these two characters. Then the distance transformation is used to classify whether the handwriting belongs to the same or different character based on the main distances.

II.MODES IN CHARACTER RECOGNITION

The basic mode of the system is given below. This system is divided in two parts.

One is training mode and other is recognition mode. During training mode database is prepared and
store the results. And during recognition mode sample character is compared with stored patterns in database and computes the result. The input data is given to the system by mouse by drawing character in Paint window or by other form, but must be in digital form.

The overall analysis can be divided into four stages:

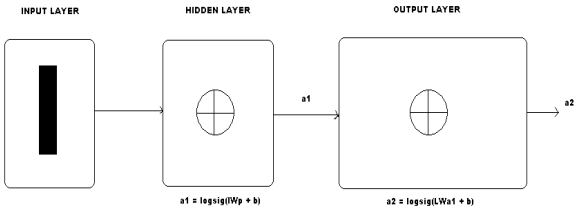
1. Image acquisition for database preparation
2. Preprocessing
3. Feature extraction, and
4. Identifying character that is recognition.

In all writers were asked to write training alphabets with variation for each alphabet and some testing character through the application interface. The input image is captured from the graphic tablet. At this stage, our system converts image into a 0-1 binary image.

A binary image composes of a collection of pixels that are 0 for the background or 1 for the foreground, arranged in a two dimensional matrix.

III.NEURAL NETWORK

The network receives the 35 Boolean values as a 35-element input vector. It is then required to identify the letter by responding with a 26-element output vector.



The 26 elements of the output vector each represent a letter. To operate correctly, the network should respond with a 1 in the position of the letter being presented to the network. All other values in the output vector should be 0.

In addition, the network should be able to handle noise. In practice, the network does not receive a perfect Boolean vector as input. Specifically, the network should make as few mistakes as possible when classifying vectors with noise of mean 0 and standard deviation of 0.2 or less.

Architecture

The neural network needs 35 inputs and 26 neurons in its output layer to identify the letters. The network is a two-layer log-sigmoid/log-sigmoid network. The log-sigmoid transfer function was picked because its output range (0 to 1) is perfect for learning to output Boolean values.

The hidden (first) layer has 10 neurons. This number was picked by guesswork and experience. If the network has trouble learning, then neurons can be added to this layer.

The network is trained to output a 1 in the correct position of the output vector and to fill the rest of the output vector with 0's. However, noisy input vectors can result in the network's not creating perfect 1's and 0's. After the network is trained the output is passed through the competitive transfer function compet. This makes sure that the output corresponding to the letter most like the noisy input vector takes on a value of 1, and all others have a value of 0. The result of this postprocessing is the output that is actually used.

Initialization

Create the two-layer network with newff.

```
net = newff(alphabet,targets,10,{'logsig','logsig'});
```

Training

To create a network that can handle noisy input vectors, it is best to train the network on both ideal and noisy vectors. To do this, the network is first trained on ideal vectors until it has a low sum squared error.

Then the network is trained on 10 sets of ideal and noisy vectors. The network is trained on two copies of the noise-free alphabet at the same time as it is trained on noisy vectors. The two copies of the noise-free alphabet are used to maintain the network's ability to classify ideal input vectors.

Unfortunately, after the training described above the network might have learned to classify some difficult noisy vectors at the expense of properly classifying a noise-free vector. Therefore, the network is again trained on just ideal vectors. This ensures that the network responds perfectly when presented with an ideal letter.

All training is done using backpropagation with both adaptive learning rate and momentum, with the function trainbpx.

IV.USE OF MATLAB WORKING COMMAND

Compare

Compare model output and measured output

Syntax

```
compare(data,m);
compare(data,m,k)
compare(data,m,k,'Samples',sampnr,'InitialState',init,'
OutputPlots
',Yplots)
compare(data,m1,m2,...,mN)
compare(data,m1,'PlotStyle1',...,mN,'PlotStyleN')
[yh,fit,x0]=
compare(data,m1,'PlotStyle1',...,mN,'PlotStyleN',k)
```

Description

Data is the output-input data in the usual iddata object format. data can also be an idfrd object with frequency-response data. Compare computes the output yh that results when the model m is simulated with the input u. m can be any idmodel or idnlmodel model object. The result is plotted together with the corresponding measured output y. The percentage of the output variation that is explained by the model

fit = 100*(1 - norm(yh - y)/norm(y-mean(y)))

is also computed and displayed. For multiple-output systems, this is done separately for each output. For frequency-domain data (or in general for complex valued data) the fit is still calculated as above, but only the absolute values of y and yh are plotted.

When the argument k is specified, the k step-ahead prediction of y according to the model m are computed instead of the simulated output. In the calculation of , the model can use outputs up to time : , , … (and inputs up to the current time t).

The default value of k is inf, which gives a pure simulation from the input only. Note that for frequency-domain data, only simulation (k = inf) is allowed, and for time-series data (no input) only prediction (k not inf) is possible.

Property Name/Property Value Pairs

The optional property name/property value pairs 'Samples'/sampnr, 'InitialState'/init, and 'OutputPlots'/Yplots can be given in any order.The argument Yplots can be a cell array of strings.

Only the outputs with OutputName in this array are plotted, while all are used for the necessary computations. If Yplots is not specified, all outputs are plotted. The argument sampnr indicates that only the sample numbers in this row vector are plotted and

used for the calculation of the fit. The whole data record is used for the simulation/prediction.
The argument init determines how to handle initial conditions in the models:
init = 'e' (for 'estimate') estimates the initial conditions for best fit.
init = 'm' (for 'model') used the model's internally stored initial state.
init = 'z' (for 'zero') uses zero initial conditions.
init = x0, where x0 is a column vector of the same size as the state vector of the models, uses x0 as the initial state.
init = 'e' is the default.

Several Models

When several models are specified, as in compare(data,m1,m2,...,mN), the plots show responses and fits for all models. In that case data should contain all inputs and outputs that are required for the different models.

However, some models might correspond to subselections of channels and might not need all channels in data. In that case the proper handling of signals is based on the InputNames and OutputNames of data and the models.
With
compare(data,m1,'PlotStyle1',...mN,'PlotStyle2'), the color, line style, and/or marker can be specified for the curves associated with the different models. The markers are the same as for the regular plot command. For example,
compare(data,m1,'g_*',m2,'r:')
If data contains several experiments, separate plots are given for the different experiments. In this case sampnr, if specified, must be a cell array with as many entries as there are experiments.
Arguments
When output arguments [yh,fit,x0] = compare(data,m1,..,mN) are specified, no plots are produced.
yh is a cell array of length equal to the number of models. Each cell contains the corresponding model output as an iddata object.
fit is, in the general case, a 3-D array with fit(kexp,kmod,ky) containing the fit (computed as above) for output ky, model kmod, and experiment kexp.
x0 is a cell array, such that x0{kmod} is the estimated initial state for model number kmod. If data is multiexperiment, X0{kmod} is a matrix whose column number kexp is the initial state vector for experiment number kexp.
Examples
Split the data record into two parts. Use the first one for estimating a model and the second one to check the model's ability to predict six steps ahead.

ze = z(1:250);
zv = z(251:500);
m= armax(ze,[2 3 1 0]);
compare(zv,m,6);
compare(zv,m,6,'Init','z') % No estimation of
% the initial state.

V. APPLICATIONS OF CHARACTER RECOGNITION

This thesis describes an algorithm that attempts to work with a subset of the features in a character that a human would typically see for the identification of machine-printed Devnagri characters. Its recognition rate is as high as the recognition rates of the older, more developed character recognition algorithms, but it is expected that if it were expanded to work with a larger set of features this could be more advanced recognizer. If it were expanded to use more features, it would be made correspondingly slower; with the advent of faster microprocessors this fact is not viewed as a crippling problem. Another research area that is receiving a lot of attention now a day is the area of pattern recognition techniques for personal identification. Pattern recognition techniques can be used to protect PCs and networks from unauthorized access by authenticating user's base along with some other physical feature such as a fingerprint, retina, iris, hand, or face. Although voice and signature identification do not involve physical characteristics, they are usually included with pattern recognition techniques.

Hence the area of application for handwritten character recognition is as given below, • In ministry of documentary analysis.

- License checking in RTO office.
- Passport verification in airline services.
- Signature analysis in banking.
- Various field of handwritten word recognition.
- Handwriting Analysis Environment.

VI. CONCLUSION

The above-discussed method has all the characteristics that are required for an off-line handwritten character system. That is simplicity and shortness of calculations. The method was found to be extremely reliable in preliminary dataset. The method can easily be applied to any application that requires handwritten character recognition, regardless of its computing power.

This is due to the low computational requirement. Thus, the proposed algorithm can be implemented on any type of software platform. The

method can also be applied to an on-line system if the coordinate data sent into the system can be sent in as a time ordered sequence of data.

Despite variations in character size, orientation, and position, the pattern recognizer system was still able to recognize many of the characters. While 2D image recognition is only part of the solution pattern recognition can bring to handwriting character recognition. Combined with Euclidean distance analysis and temporal information, pattern recognition look to be a very promising solution. The results of this project are hardly new. While pattern recognition is a promising solution there are some short-term problems. Conducting experiments on this project, it became clear that correctly training character for database; it can be a very time and processor intensive activity. This has resulted in some researchers advocating 'lazy recognition' that is offline recognition, not attempting to do character recognition in real time. Microsoft has also adopted a strategy with its upcoming Tablet PC to do character recognition silently in the background instead of in real time.

REFERENCES

[1] N. Sharma, U. Pal, F. Kimura, S. pal, "Recognition of Off Line Handwritten Devnagari Characters using Quadratic Classifier", ICCGIP 2006, LNCS 4338, 2006.

[2] R. Kapoor, D. Bagai and T.S. Kamal, "Representation and Extraction of Nodal Features of DevNagri Letters", Proceedings of the 3rd Indian Conference on Computer Vision, Graphics and Image Processing.

[3] Reena Bajaj, Lipika Dey, and S. Chaudhury, "Devnagari numeral recognition by combining decision of multiple connectionist classifiers",Sadhana, Vol.27, part. 1, , 2002

[4] Bellili, M. Gilloux, P. Gallinari, An MLP-SVM combination architecture for o²ine handwritten digit recognition: reduction of recognition errors by support vector machines rejection mechanisms, Int. J. Document Analysis and Recognition, 5(4): 2003.

[5] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. System Man Cybernet.,22, 1992.

[6] S. Arora, D. Bhattacharjee, M. Nasipuri, M.Kundu, D.K. Basu, "Application of Statistical Features in Handwritten Devnagari Character Recognition", International Journal of Recent Trends in Engineering[ISSN 1797-9617], IJRTE Nov 2009

[7] U. Kreßel, J. Schurmann: Pattern classification techniques based on function approximation. In: H. Bunke,P.S.P. Wang (eds.), Handbook of Character Recognition and Document Image Analysis, World Scientific, Singapore, 1997.

[8] R2012a Documentation(Matlab)

[9] A FEATURE EXTRACTION TECHNIQUE BASED ON CHARACTER GEOMETRY FOR CHARACTER RECOGNITION By Dinesh Dileep